NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A GRAPH THEORETIC APPROACH TO THE OPTIMAL SLOT UTILIZATION PROBLEM FOR NAVAL COMMUNICATION NETWORKS

by

Pamela K. Bell

June, 1992

Thesis Advisor: Craig W. Rasmussen

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL *(If applicable)* 30	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000	7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS Program Element No. / Project No. / Task No. / Work Unit Accession Number

11. TITLE *(Include Security Classification)*
A GRAPH THEORETIC APPROACH TO THE OPTIMAL SLOT UTILIZATION PROBLEM FOR NAVAL COMMUNICATION NETWORKS

12. PERSONAL AUTHOR(S) Bell, Pamela K.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(year, month, day)*	15. PAGE COUNT
Master's Thesis	From To	1992, June	60

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUBGROUP	Graphs, Graph Coloring, Conflict Graphs, Chromatic Numbers

19. ABSTRACT *(continue on reverse if necessary and identify by block number)*

This paper approaches the optimal slot utilization problem in a Naval Battle Group by modelling ships capable of transmitting on a particular frequency as vertices in a graph, and the relationships between them as edges in that graph. We then analyze the structure of the resultant graph and find an upper bound on the chromatic number of its conflict graph to take into account all possible patterns of interference in determining the minimum number of time slots required, thereby allowing efficient and effective net throughput. Our results include the identification of specific types of graphs in which an exact solution is possible based upon the maximum degree of all vertices in the graph, as well as an algorithm for general graphs which will identify an upper bound on the chromatic number of their conflict graph. Original results are proven, and analysis and examples of the algorithm are provided.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Craig W. Rasmussen	22b. TELEPHONE *(Include Area code)* (408) 646-3409	22c. OFFICE SYMBOL MA/Ra

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

A Graph Theoretic Approach to the Optimal Slot Utilization Problem for Naval Communication Networks

by

Pamela K. Bell
Lieutenant, United States Navy
B.A., SUNY College at Potsdam, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS
from the

NAVAL POSTGRADUATE SCHOOL
June 1992

Author: ______________________
Pamela K. Bell

Approved by: ______________________
Craig W. Rasmussen, Thesis Advisor

____
John R. Thornton, Second Reader

____
Richard H. Franke, Chairman
Department of Mathematics

ABSTRACT

This paper approaches the optimal slot utilization problem in a Naval Battle Group by modelling ships capable of transmitting on a particular frequency as vertices in a graph, and the relationships between them as edges in that graph. We then analyze the structure of the resultant graph and find an upper bound on the chromatic number of its conflict graph to take into account all possible patterns of interference in determining the minimum number of time slots required, thereby allowing efficient and effective net throughput. Our results include the identification of specific types of graphs in which an exact solution is possible based upon the maximum degree of all vertices in the graph, as well as an algorithm for general graphs which will identify an upper bound on the chromatic number of their conflict graph. Original results are proven, and analysis and examples of the algorithm are provided.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. STATEMENT OF PROBLEM

The **Shared Adaptive Internetworking Technology** (SAINT) Project has been undertaken by the Naval Oceanographic Systems Center (NOSC) to address advanced concepts in fleet communication networking, focusing on the development of new and improved protocols for Navy Communication Subnetworks, multiband selection, and Military Enhanced OSI Protocols. [Ref. 1: p. 3]

The specific area of interest for this thesis is that of Communication Subnetworks and the concept of Improved Slot Utilization. A slot, in this context, is a time unit in a message transmission cycle on a specific frequency during which a given ship is authorized to transmit messages to other ships in the Battle Group. The current protocol allocates a set of slots to each ship in the Battle Group without regard to possible nonconflicting transmissions. It is therefore desirable to enhance the current protocol to allow increased utilization of slots, utilizing connectivity information to allow users in separate parts of the network to share the same slot, thus improving net throughput. [Ref. 1: p. 7]

An example of possible slot allocations for a Battle Group is depicted in Figure 1.1.

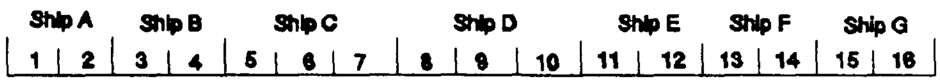


Figure 1.1 Slot Assignments for a Battle Group

The basic premise of this thesis is that if two ships are able to transmit simultaneously to different receiving ships without conflicting in their transmissions, then the assignment of a single time slot to both ships should be possible.

The mathematical model we will be using to examine the slot utilization problem for a given Battle Group is a graph, where the ships are represented by vertices and the capability of two ships to communicate between one another on a given frequency is represented by an edge between their respective vertices. Now, by analyzing certain characteristics of the graph, we hope to be able to give specific results as to an optimal slot utilization scheme.

Figure 1.2 shows the graphical representation of the Battle Group shown in Figure 1.1, and gives an example of slot assignments which provide improved slot utilization.

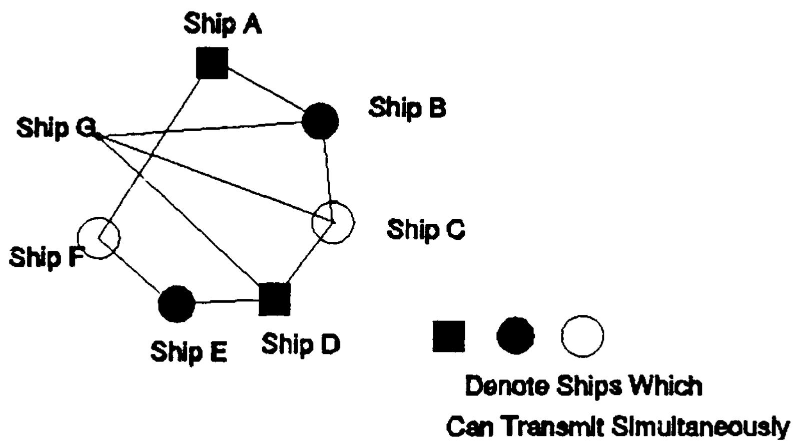


Figure 1.2 Improved Slot Utilization

The remainder of Chapter I provides basic definitions in Graph Theory to be used throughout this text as well as some specific classes of graphs and their identifying characteristics. To determine the interference relationships between ships in the Battle Group, and hence vertices in the graphical model, we will be using concepts in constructing conflict graphs as introduced in Chapter II. We will then identify the idea of graph coloring as the mathematical technique to be used in selecting ships which may use the same time slot in the transmission cycle without encountering any interference. Chapter III continues with an introduction to coloring algorithms and the theory used to analyze those algorithms in terms of efficiency and running time.

Finally, Chapter IV introduces a number of original results developed for specific classes of graphs that have an optimal upper bound on the number of slots required. We then consider general graphs and introduce an original algorithm to obtain appropriate results for the general graphs efficiently. We complete the paper with a chapter on summary and conclusions, and a discussion on how these results may be used in the original slot optimization problem.

B. GRAPH THEORY

1. Background

Persons familiar with the fundamentals of graph theory may wish to overlook this section and continue their reading with Chapter II.

Graphs were first introduced by Euler in 1736 in his discussion of the now famous Konigsberg Bridge Problem. The problem is explained that in the old city of Konigsberg in East Prussia, the city was located along the banks and two islands of the Pregel river with the four parts of the city connected by 7 bridges as shown in Figure 1.3. On Sundays, the citizens of Konigsberg would promenade about town, and the problem arose as to whether it was possible to plan a promenade so that starting at home one returns there after having crossed each of the 7 bridges exactly once. [Ref. 2: p. p. 415] Euler studied the problem and deduced that indeed it

was impossible to complete the promenade with the city arranged as such.

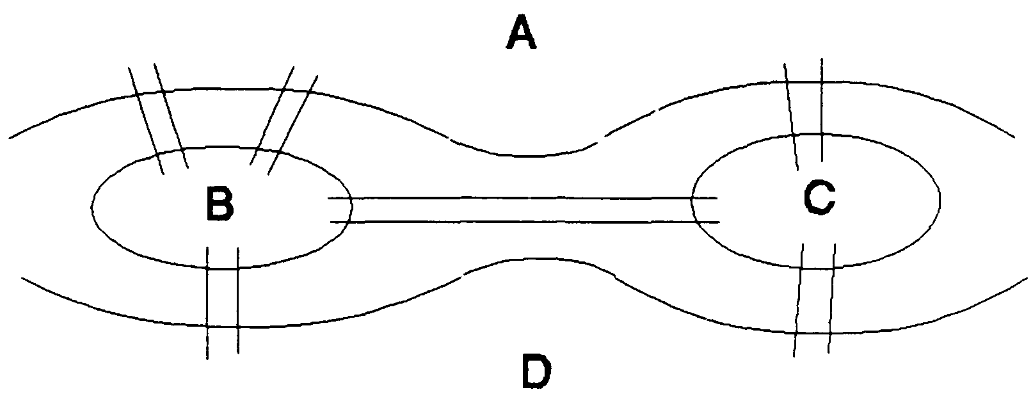


Figure 1.3 Konigsberg Bridge Problem

In doing so, he replaced the map of Konigsberg with the graph drawn in Figure 1.4, and redefined the problem in present terminology as whether there exists a closed path which contains all the edges of the graph. [Ref. 2: p. 416]

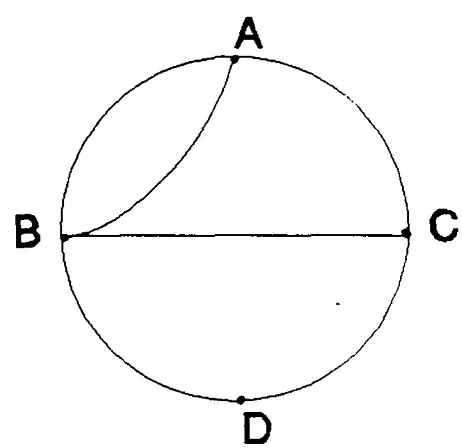


Figure 1.4 Graphical Representation of Konigsberg Bridge Problem

2. Definitions and Terminology

The initial definitions in basic graph theory have been adopted from Fred S. Roberts [Ref. 3: p. 82-91].

A digraph D is a pair (V,A), where V is a set of p vertices and A is a set of ordered pairs of elements of V. The vertices are represented by points and there is a directed arc heading from u to v if and only if (u,v) is in A.

In Figure 1.5, V is the set {a,b,c,d,e} and A is the set {(a,b),(a,d),(b,c),(b,d),(c,a),(c,d),(d,c),(d,e),(e,a),(e,b)}.

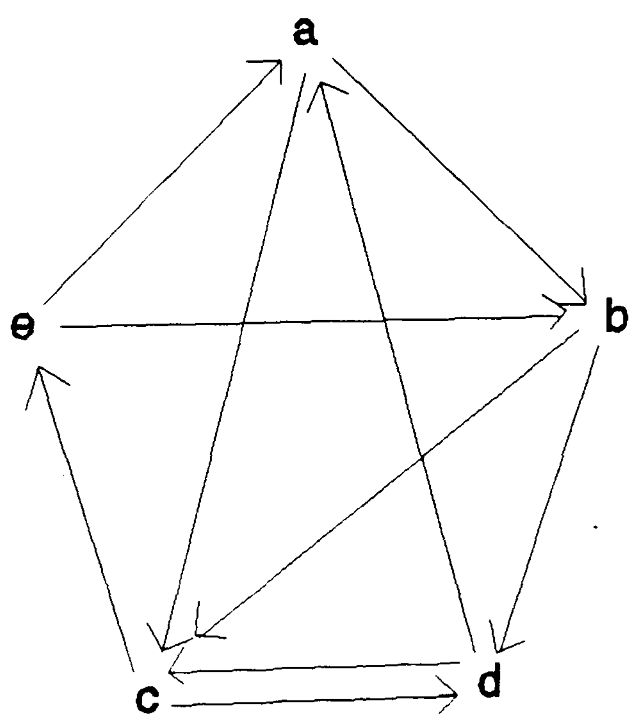


Figure 1.5 A Digraph

A graph G is a pair (V,E), where V is a set of p vertices and E is a set of q unordered pairs of elements of V.

The graph G is said to have order p and size q. Again, the vertices are represented by points, while the edges between two points u and v are drawn if and only if {u,v} is in E. Figure 1.6 shows three simple examples of undirected graphs.

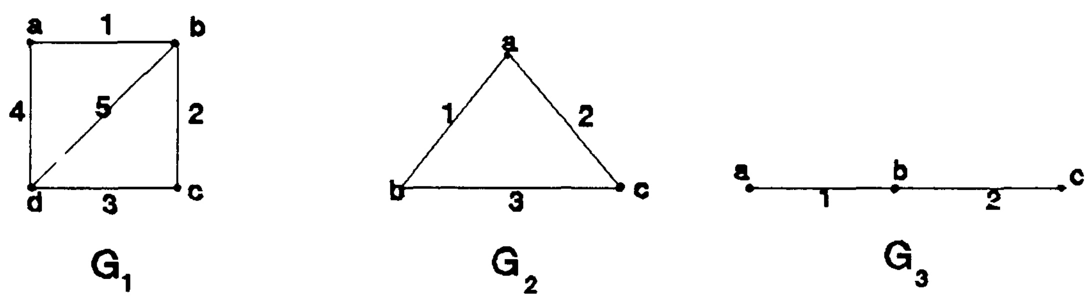


Figure 1.6 Examples of Graphs

In Figure 1.6, graph G_1 consists of V={a,b,c,d} and E={{a,b},{b,c},{c,d},{a,d},{b,d}}, while G_2 consists of V={a,b,c} and E={{a,b},{b,c},{a,c}}.

When there is an edge between vertex u and vertex v, we say u is <u>adjacent to</u> v, and similarly v is adjacent to u. Further, an edge which lies between two vertices, say u and v, is said to be <u>incident to</u> vertex u and incident to vertex v. In our examples above, in graph G_1, a and b are adjacent vertices and edge 1 is incident to vertex a and vertex b. The neighborhood of a vertex is defined to be the set of vertices which are adjacent to the vertex.

We will assume the following about the graphs used in this study. There are no multiple edges, i.e., no more than

one edge from one vertex to another, and no loops, i.e., edges from a vertex to itself.

The degree of a vertex is the number of edges incident with it and is denoted deg(v). In G_2 of Figure 1.6, deg(a)=2, while in G_1, deg(b)=3. The degree sequence of a graph is a list of the degrees of the vertices in nonincreasing order. The minimum degree among the vertices of a graph G is denoted $\delta(G)$, while the maximum degree is denoted $\Delta(G)$.

C. DISTANCE IN GRAPHS

There is much emphasis placed upon the concept of distance in our graphs throughout this study. The basic definitions used in this section have been adopted from F. Harary and F. Buckley [Ref. 4: p. 2-15].

A walk in a graph G is an alternating sequence of vertices and edges $v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$, such that every $e_i = v_{i-1}v_i$ is an edge of G, $1 \le i \le n$. The vertices and edges need not be distinct. A walk is a path if all of its vertices (and hence its edges) are distinct.

The distance d(u,v) between two vertices u and v in G is the minimum number of edges in a path joining them, if any. A graph is connected is there is a path joining each pair of vertices.

In a connected graph G, the eccentricity of a vertex v, denoted e(v), is the distance to a vertex farthest from v. Thus $e(v) = \max\{d(u,v) : v \in V\}$. The radius of a graph G, denoted

r(G), is the minimum eccentricity of any vertex in G and the diameter of a graph G, denoted d(G), is the maximum such eccentricity. [Ref. 4: p. 31-2]

D. TYPES OF GRAPHS

There are several types of graphs that will be used as examples later in this paper, grouped according to specific characteristics they have in common such as diameter, radius, or degree sequences.

A regular or k-regular graph is a graph in which all vertices have the same degree k. A graph with p vertices is said to be complete, denoted K_p, when every pair of its vertices is adjacent. In Figure 1.7, graph G_1 is a 3-regular graph and graph G_2 is a complete graph K_5.

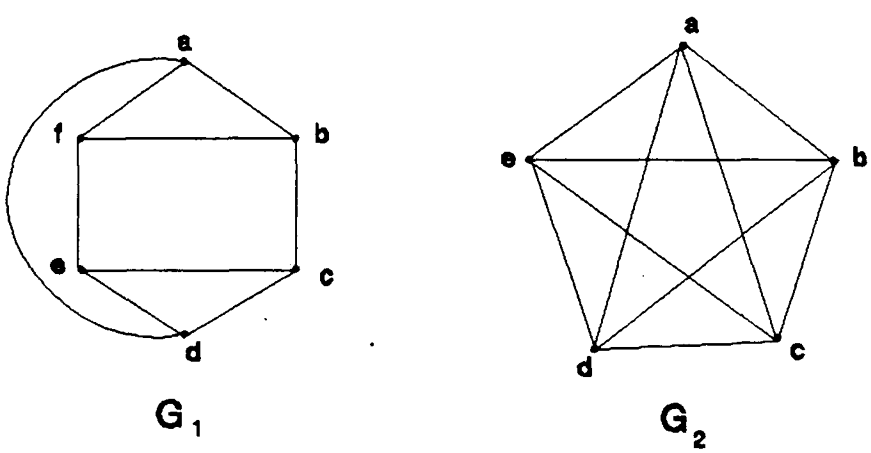


Figure 1.7 Examples of a Regular Graph and Complete Graph

A graph G is bipartite if the vertex set V can be partitioned into two subsets V_1 and V_2, such that every edge in G joins a vertex in V_1 with a vertex in V_2. If G is bipartite

and contains every possible edge joining V_1 and V_2, then G is a complete bipartite graph. If V_1 and V_2 have m and n vertices, we write $G=K_{m,n}$. A tree is a special type of a bipartite graph where every two vertices are joined by a unique path and there are no cycles. In Figure 1.8, graph G_1 is a bipartite graph in which $V_1 = \{a',b',c'\}$ and $V_2 = \{a,b\}$, G_2 is a complete bipartite graph in which $V_1 = \{u',v',w',x'\}$ and $V_2 = \{u,v,w\}$, and G_3 is a tree where we may partition the vertices as $V_1 = \{a,d,e,f\}$ and $V_2 = \{b,c,g,h,i,j\}$.

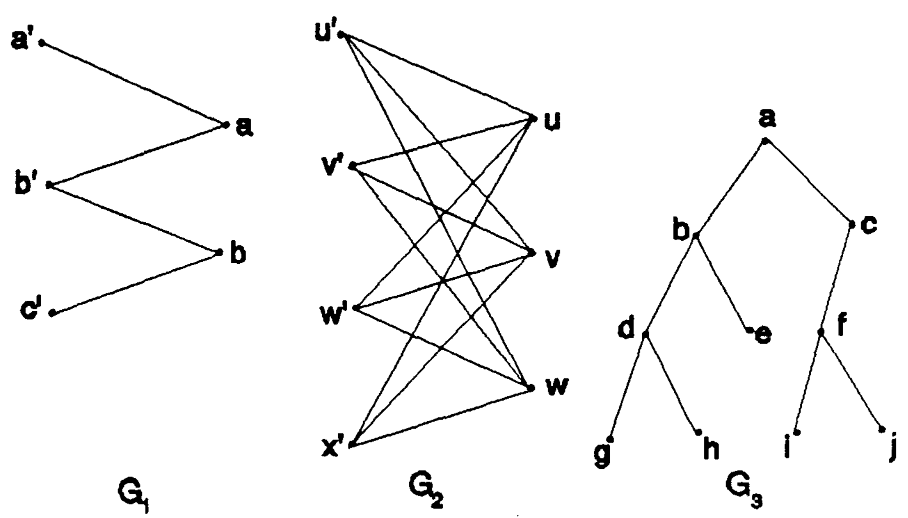


Figure 1.8 A Bipartite Graph, Complete Bipartite Graph, and a Tree

A cycle is a closed walk in which all of the vertices except the starting and ending vertices are distinct. We denote a cycle of length p as C_p. Given two graphs G_1 and G_2 with distinct node sets V_1 and V_2, and edge sets E_1 and E_2, respectively, a join is denoted $G_1 + G_2$, where $V = V_1 \bigcup V_2$ and

$E = E_1 \bigcup E_2 \bigcup \{\{x,y\} \mid x \varepsilon V_1,\ y \varepsilon V_2\}$. A wheel, denoted $W_{m,n}$ is formed by the join of a complete graph K_m and a cycle C_n. In Figure 1.9, graph G_1 is a cycle C_5 while G_2 is a wheel $W_{1,5} = K_1 + C_5$.

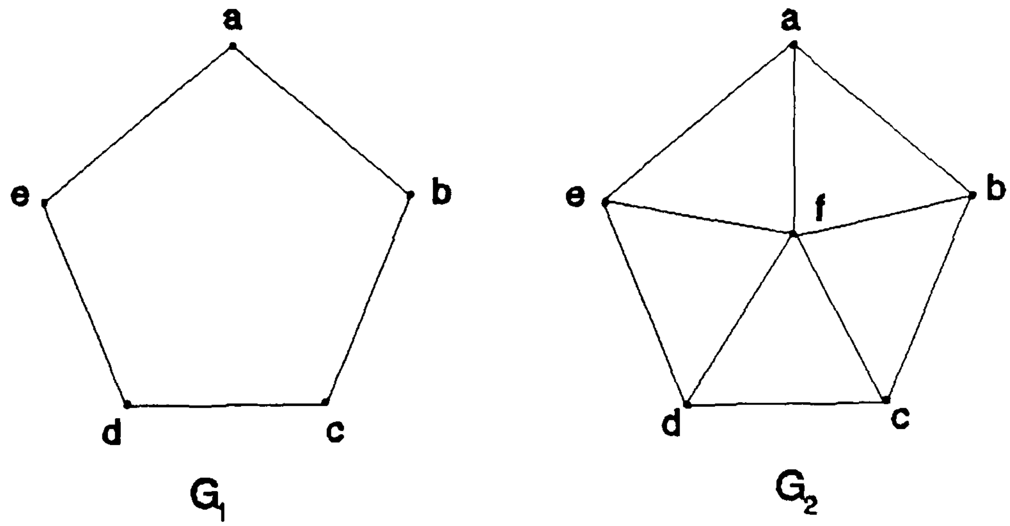


Figure 1.9 A Cycle and a Wheel

A subgraph of a graph G is a graph having all of its nodes and edges in G. It is called a spanning subgraph if it contains all the nodes of G. For any set S of nodes in G, the induced subgraph <S> is the maximal subgraph with node set S. [Ref. 4: p. 9] A complete subgraph is called a clique. The maximum order of a complete subgraph of G is called the clique number of G and is denoted $\omega(G)$. [Ref. 5: p. 244]

Suppose that G is a graph and $u_1, u_2, \ldots, u_{t-1}, u_t = u_1$ is a cycle in G. A chord in this cycle is an edge $\{u_i, u_j\}$, $i \neq j$, which appears as an edge in G. A triangular chord is a chord (u_i, u_j) such that i and j differ by 2. [Ref. 6: p. 109]

An undirected graph G is called <u>chordal</u> (triangulated) if every cycle of length strictly greater than three possesses a chord, that is, an edge joining two nonconsecutive vertices of of the cycle [Ref. 7,p. 81]. Figure 1.10 shows an example of a chordal graph and a non-chordal graph.

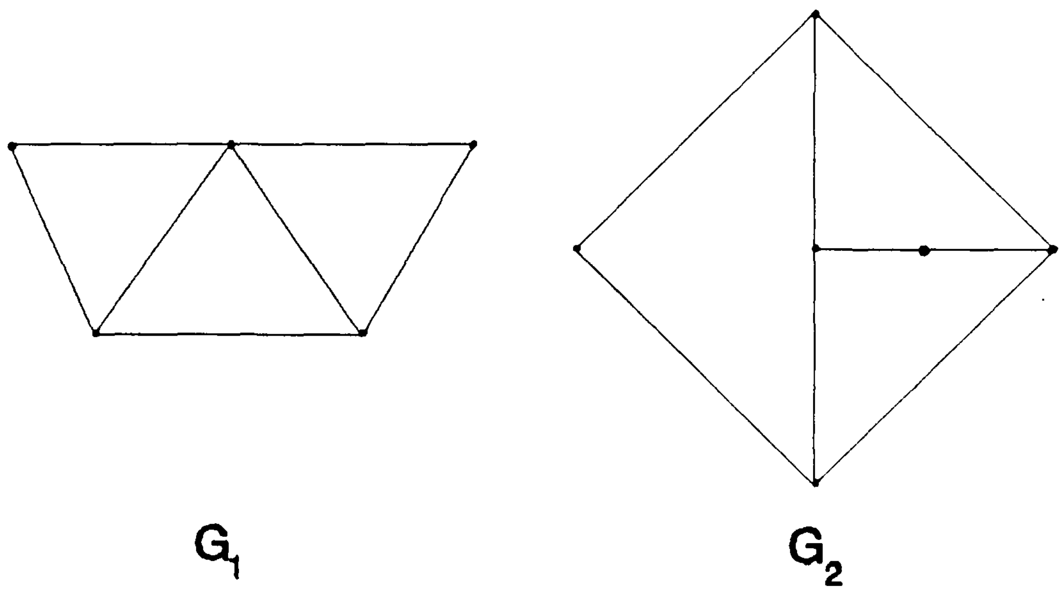


Figure 1.10 A Chordal and Non-Chordal Graph

A <u>diamond</u> is a graph consisting of two triangles with exactly one pair of vertices common to both, i.e., a complete graph on four vertices with one edge deleted. If a graph G contains no induced diamond, we call G diamond-free. Figure

1.11 shows an example of a diamond and diamond-free graph. [Ref. 8: p. 2]

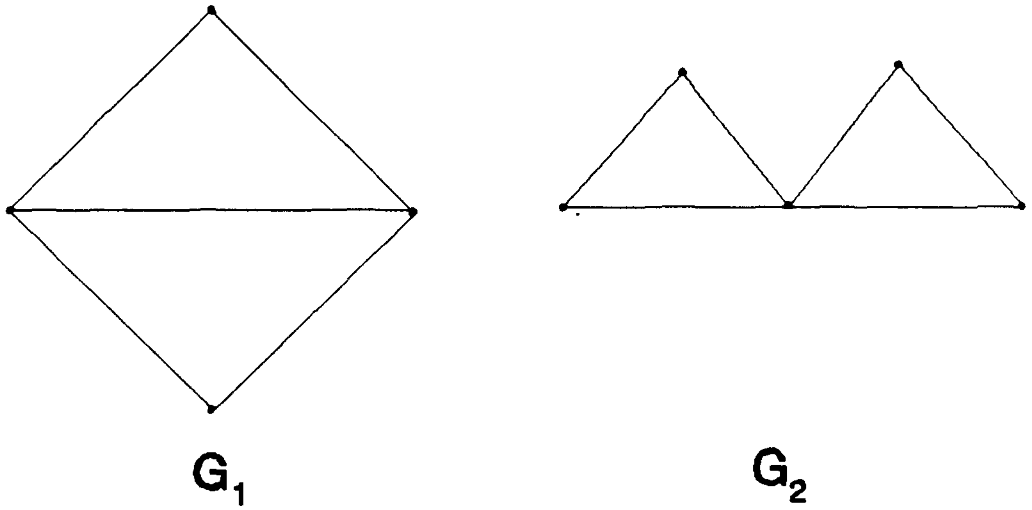


Figure 1.11 A Diamond and a Diamond-Free Graph

Thus, a <u>diamond-free chordal</u> graph is a diamond-free graph in which every cycle of length strictly greater than three possesses an edge joining two nonconsecutive vertices of the cycle.

II. COMPETITION AND CONFLICT GRAPHS

A. COMPETITION GRAPHS

1. Background

Competition Graphs were first studied by Cohen in 1968 in his analysis of food web models of ecological systems. Cohen introduced competition graphs as a means to understand the relationships between predators and prey throughout the species found in the system. Using the previously defined terms of digraphs in modelling the food web, the vertices represent species and there is an arc from species x to species y if x preys on y. Figure 2.1 shows an example of a food web. [Ref. 9: p. 1]

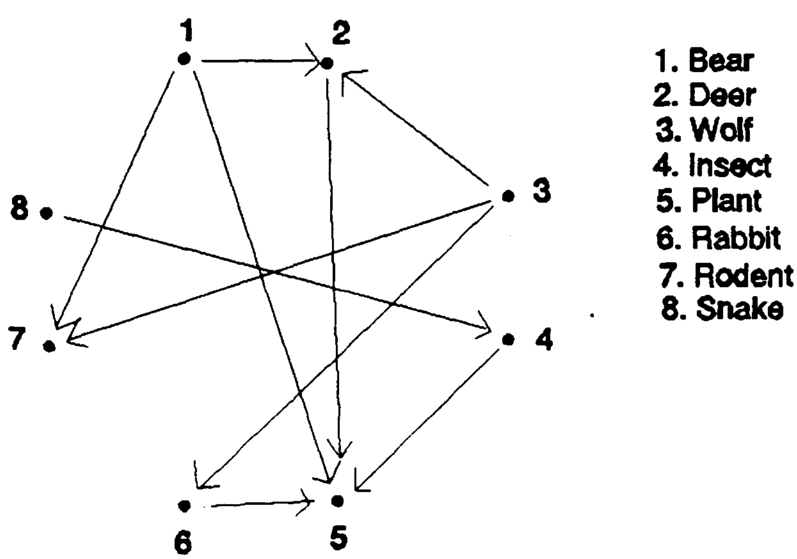


Figure 2.1 A Food Web

2. Definitions

For each digraph D=(V,A) there is a corresponding undirected graph , called its <u>competition</u> graph with vertex set equal to the original vertex set V(D), but with an undirected edge between two distinct vertices x and y of V(D) if and only if there is a vertex a such that (y,a) and (x,a) are both arcs in A(D), i.e., if and only if x and y are in competition in the sense of preying on a common species. Figure 2.2 shows a digraph D and its competition graph C(D). [Ref. 10: p. 295]

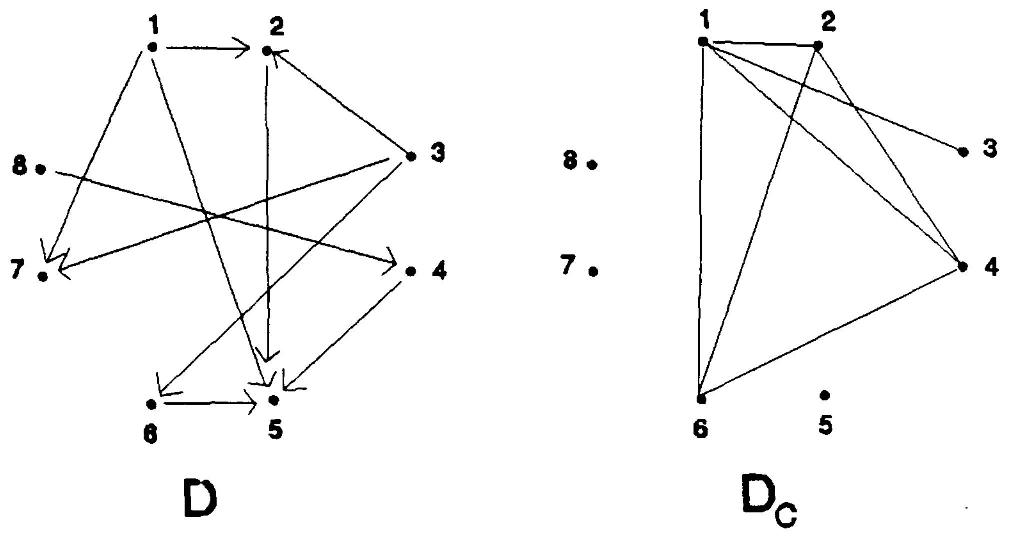


Figure 2.2 The Food Web from Figure 2.1 and Its Competition Graph

We can generalize the use of competition graphs if it is possible to partition the vertex set A into two sets B and

C, where B is the set of transmitting vertices and C is the set of receiving vertices and $B \bigcup C = A$. We may then denote the graph as G(D,B,C), where D is the digraph representing the system and B and C are defined as above.

3. Applications

Raychaudhuri and Roberts continued the study of generalized competition graphs in a number of applications. One of the more relevant examples to this thesis is that of radio or television transmitters, where we have a set B of transmitting stations and a set C of receiving stations, and define a digraph D by taking V(D) to be $B \bigcup C$ and taking an arc from x in B to a in C if a signal sent at x can be received at a. Then the graph G(D,B,C) is a special form of the competition graph called a <u>conflict</u> graph. Two transmitting stations are said to conflict if and only if a message sent by them can be received at the same place. [Ref. 10: p. 296]

B. CONFLICT GRAPHS

1. Definitions

A special form of a competition graph is the <u>conflict</u> graph, denoted G_c, which is obtained from a given undirected graph G=(V,E) in which $V(G_c)=V(G)$ and in which the edge set $E(G_c)$ consists of all edges in the original set E(G) together with all edges {u,v}, where {u,a} and {v,a} are in E(G) for some vertex a in V(G). That is, all vertices within distance

two of one another in the original graph G have an edge between them in its conflict graph. For example, in Figure 2.3 the graph G consists of V(G)={a,b,c,d,e} and E(G)={{a,b},{b,c}, {c,d},{d,e},{a,e}}, while in the conflict graph G_c, V(G_c)=V(G)= {a,b,c,d,e} and E(G_c)={{a,b},{b,c},{c,d}, {d,e},{a,e},{a,c}, {a,d},{b,d},{b,e},{c,e}}.

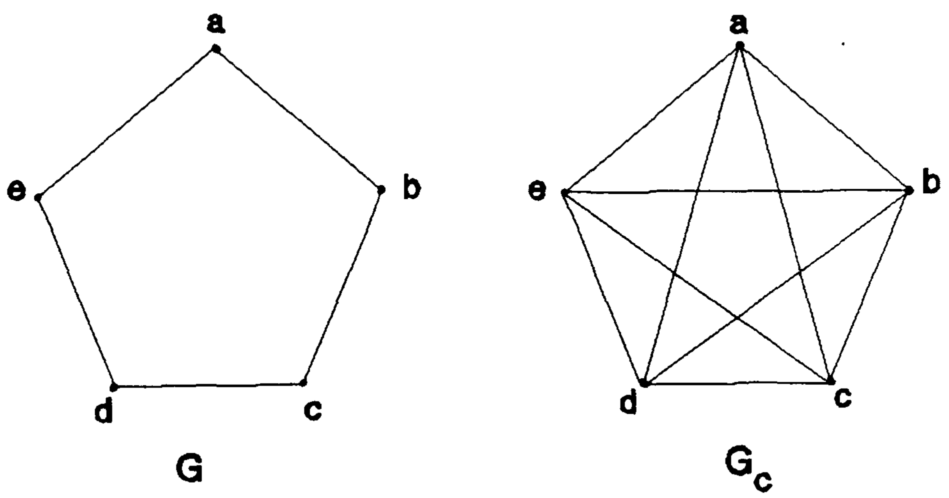


Figure 2.3 A Graph and Its Conflict Graph

Conflict graphs have proven most useful in the study of communication networks and the design of computer networking schemes.

III. GRAPH COLORING AND NP-COMPLETENESS

A. GRAPH COLORING

1. Background

The problem of graph coloring was first introduced in 1852 by Francis Guthrie and Augustus De Morgan. While they could not prove it, both mathematicians conjectured that any map drawn in the plane could be colored with at most four colors. [Ref. 5: p. 239] Figure 3.1 shows an example of a map coloring with six countries. [Ref. 3: p. 100]

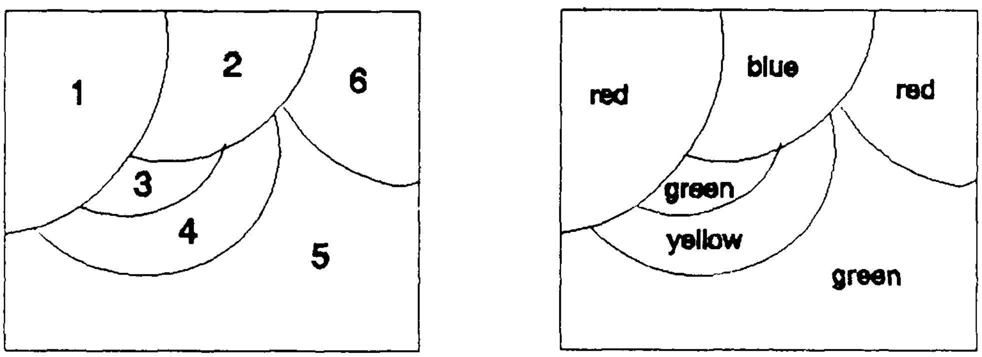


Figure 3.1 A Map and Its Coloring

In the map above, Country 1 was colored red. Given that Country 2 shares a common boundary with Country 1, we must color Country 2 with a new color, say blue. Now Country

3 shares boundaries with both Country 1 and Country 2, so we must use another color, say green. Similarly, Country 4 shares boundaries with each of Countries 1, 2 and 3, so we may introduce a fourth color, say yellow. Now Country 5 shares a boundary with Countries 1, 2 and 4, but is not adjacent to Country 3 and may therefore use the color green. And finally, Country 6 shares a common boundary with Countries 2 and 3, but is not adjacent to Country 1 and may therefore use the color red.

While Guthrie and De Morgan's Four-Color conjecture remained unproven for over 100 years, it was finally proven in 1977 by Appel and Haken. [Ref. 3: p. 100]

2. Definitions and Terminology

In properly coloring a graph, we assign a color to each vertex of a graph G in such a way that if two vertices are joined by an edge, they get a different color. If such an assignment can be carried out using at most k colors, we call it a <u>k-coloring</u> of G and say G is <u>k-colorable</u>. The smallest number k such that G is k-colorable is called the <u>chromatic number</u> of G and is denoted by $\chi(G)$ [Ref. 3: p. 97-8]. Figure 3.2 shows two examples of graph colorings in graphs G_1 and G_2. Clearly, if a graph G has n vertices each vertex could be colored distinctly, but the challenge lies in finding the chromatic number or bounds upon the value of the chromatic number.

There are several algorithms available for obtaining bounds on the chromatic number of a graph, however none are able to guarantee the exact calculation of its value.

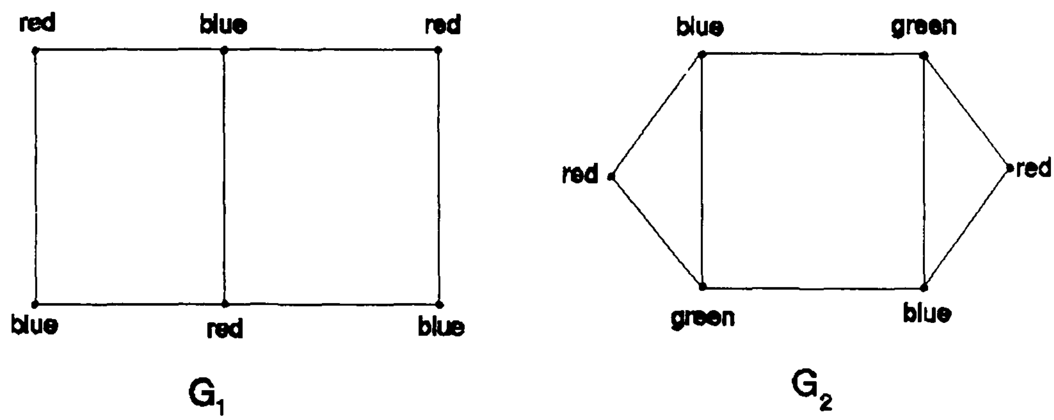


Figure 3.2 Examples of Graph Colorings

3. Applications

The four-color problem provided a stepping stone to theories in graph coloring. Some interesting applications are the scheduled meeting problem, channel assignment problem, and the scheduling of garbage collection tours [Ref. 11: p. 2-4], as discussed below.

In the scheduled meetings problem, the New York State Assembly used graph coloring to solve the problem of assigning a meeting time to each legislative committee. If two committees have a member in common, they must yet a different meeting time. A graph is constructed in which the vertices

represent committees and an edge between vertices x and y exists if and only if committees x and y have a member in common. The colors used in the graph then represent meeting time blocks, and the chromatic number is the distinct smallest number of such meeting blocks required.

The channel assignment problem is used in radio and television applications where we wish to assign a channel to each television or radio transmitter. In a simplified version, transmitters interfere if and only if they are within a certain distance of each other, and hence those transmitters which interfere must get different channels. We can model this interference by constructing the conflict graph, where the vertices represent transmitters and an edge between vertices x and y exists if and only if the corresponding transmitters interfere. The colors of the graph are channels, and hence the chromatic number of the graph represents the smallest spectrum of channels required in the system.

The garbage collection problem deals with a tour of a garbage truck, or the schedule of sites it visits on a given day, with the constraint that if two tours visit the same site they must do so on different days. We wish to assign to each tour a day of the week on which it will be scheduled so that if two tours visit a given site, they get a different day. In the graph for this problem, the vertices represent the tours and an edge exists between vertices x and y if and only if tours x and y visit a common site. The colors used in the

graph then designate all tours that can be executed on the same day. Hence in order to hold to a six-day work week, we wish to find whether the graph is 6-colorable.

B. THEORY OF NP-COMPLETENESS

To appreciate the difficulties in graph coloring it is important to understand the algorithms used to construct the coloring and to analyze the behavior of the algorithms without having to implement them in every situation possible. This analysis leads to the study of the complexity of algorithms and to the issue of NP-Completeness. Readers are referred to Manber for a more complete background on the analysis of algorithms. [Ref. 12: p. 37-40]

In analyzing the behavior of an algorithm it is important to understand that different algorithms accomplish different tasks. Thus we need a method to compare competing algorithms or a way to determine the feasibility of performing the required calculations using a specific algorithm. It is important also to relate computational effort in the algorithm to the size of the problem, usually measured in terms of hte input parameters. For instance, in algorithms dealing with graphs the computational effort could be related to the number of vertices or the number of edges in the graph. It is common in practice to use worst-case scenarios in approximating these input sizes.

The complexity function of an algorithm is based upon the input size n and the number of primary steps in the algorithm as a function of n. We will use the notation that a function $g(n)$ is $O(f(n))$ for a complexity function $f(n)$ if there exist positive constants c and N such that for all $n \geq N$, we have $|g(n)| \leq c|f(n)|$. The O notation hence bounds $g(n)$ only from above. For example, $6n^2+10 = O(n^2)$, since $6n^2+10 \leq 7n^2$ for $n \geq 4$. A polynomial time algorithm is defined to be one whose time complexity function is $O(p(n))$ for some polynomial function p. Any algorithm whose time complexity function cannot be bounded by a polynomial is called an exponential time algorithm.

The theory of NP-Completeness, as discussed by Garey and Johnson [Ref.14: p. 6-8] is based upon the distinction between polynomial time algorithms and exponential time algorithms. The distinction between these two types of algorithms has particular significance when considering the solution of problems with large input size n. Table 1 illustrates the differences in growth rates among several typical complexity functions of each type, where the functions express running time. Notice the explosive growth rates for the exponential complexity functions.

The table below indicates one of the primary reasons why polynomial time algorithms are generally regarded as being more desirable than exponential time algorithms, and indeed it has been proven that polynomial time algorithms are always

preferable to exponential time algorithms. The reader is referred to Garey and Johnson for the proof and a more complete discussion on NP-Completeness.

TABLE 1. COMPARISON OF POLYNOMIAL AND EXPONENTIAL TIME COMPLEXITY FUNCTIONS [Ref. 13: p. 7]

Time Complexity Function	Size n 10	20	30	40	50
n	.00001sec	.00002sec	.00003sec	.00004sec	.00005sec
n^2	.0001sec	.0004sec	.0009sec	.0016sec	.0025sec
n^5	.1sec	3.2sec	24.3sec	1.7min	5.2min
2^n	.001sec	1.0sec	17.9min	12.7days	35.7yrs
3^n	.059sec	58min	6.5yrs	3855cent	2×10^8cent

For our purposes, it is enough to say that an NP-Complete problem is intractable for all but small problem sizes.

C. COLORING ALGORITHMS

While the general question of determining the chromatic number of a graph is known to be NP-Complete, there are a number of known polynomial-time algorithms which give approximate colorings of graphs and which will guarantee optimality in specific instances. For example, the Brelaz Color-Degree Algorithm described below determines the chromatic number for a given graph G.

Brelaz Color-Degree Algorithm

Input: A graph G=(V,E)

Output: A proper coloring of the vertices of G using colors 1, 2, ..., |V|

Method: Break the ties based on the smallest color-degree (The color-degree of a vertex v is defined to be the number of colors needed to color the vertices adjacent to v.)

1. Order the vertices in decreasing order of degrees.
2. Color a vertex of largest degree with color 1.
3. Select from the uncolored vertices a vertex with maximum color-degree. If there is a tie, choose any vertex of largest degree in the uncolored subgraph.
4. Color the vertex selected in Step 3 with the lowest possible color number.
5. If all vertices are colored, then stop; else go to Step 3.

Another well-known algorithm discussed by Golumbic [Ref. 7: p. 98-9] correctly calculates the chromatic number of a triangulated graph G. A <u>triangulated</u> graph is defined to be one in which its maximal complete subgraph is a triangle. [Ref. 7: p. 98-9]. The details of that algorithm may be found in the reference above.

Although the above algorithms can guarantee optimal colorings, the graphs that best model the battle groups introduced in Chapter I can not be categorized into the required bipartite or triangulated form. It should prove beneficial, then, to instead discuss those algorithms or

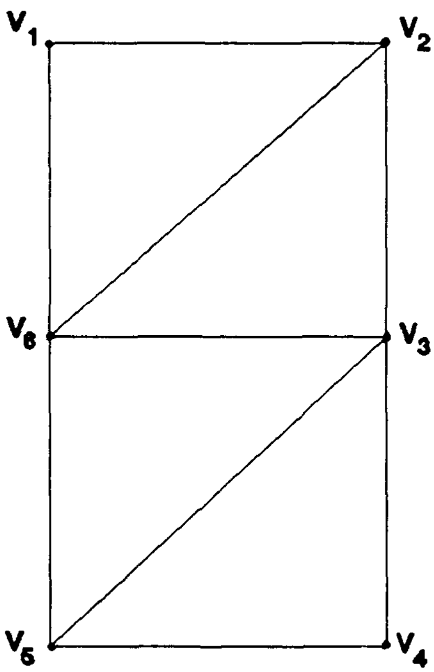
heuristics which may provide near-optimal results for the types of graphs more often encountered.

A <u>heuristic</u> is defined to be a technique that improves the efficiency of a search, while possibly sacrificing claims of completeness, or, in this case, optimality. The technique we most often use is the nearest neighbor algorithm following the best-first search process in which we examine all unvisited neighbors of some starting vertex and next visit the neighbor that most satisfies some test criterion. This is done in the hope that the test criterion will point us more rapidly toward our desired solution, rather than making it necessary to enumerate all possible solutions. [Ref. 5: p. 58]

The first heuristic normally encountered in graph coloring literature is that of the greedy graph coloring method. The idea is to color the vertices as they are encountered using any color available, that is, any color not already assigned to a neighboring vertex. This approach can lead to a very poor upper bound on the chromatic number. For example, in Figure 3.3, the graph G has been colored starting at vertex v_1 and continuing through the consecutively ordered vertices, resulting in a coloring number of 4, while the chromatic number of the graph can be shown to be 3.

It has been found that the "best" approximating algorithms are motivated by a small number of heuristics based upon the

following three ideas, first described by B. Manvel. [Ref. 5: p. 229]



Greedy Coloring	Optimal Coloring
v_1 :red	v_1 :red
v_2:blue	v_2:green
v_3:red	v_3:red
v_4:blue	v_4:blue
v_5 :green	v_5 :green
v_6:yellow	v_6:blue

Figure 3.3 A Greedy Graph Coloring and An Optimal Coloring

1. A vertex of high degree is harder to color than a vertex of low degree.
2. Vertices with the same neighborhood should be colored alike.
3. Coloring many vertices with the same color is a good idea.

One such algorithm is the Welsh-Powell Coloring Algorithm which specifies that vertices of higher degree are colored

first, then multiple vertices are colored with the same color, if possible.

Welsh-Powell Algorithm

Input: A graph G=(V,E)

Output: An approximate coloring of the vertices of G

Method: Color first uncolored vertex and uncolored vertices not adjacent to it.

1. Order the vertices in decreasing order by degree.
2. Assign an unused color to the first uncolored vertex in the list. Go through the list in order, assigning the same color to any vertex not adjacent to any other vertex with this color.
3. If some vertices are not colored, go to Step 2.
4. Done.

IV. COLORING CONFLICT GRAPHS

A. SPECIAL TYPES OF CONFLICT GRAPHS

In general, a closed form expression for the chromatic number of the conflict graph of a graph G in terms of one or more parameters of G is not obtainable. There are, however, classes of graphs whose members do allow that. For example, a tree like that pictured below in Figure 4.1, is a type of graph whose structure is so defined.

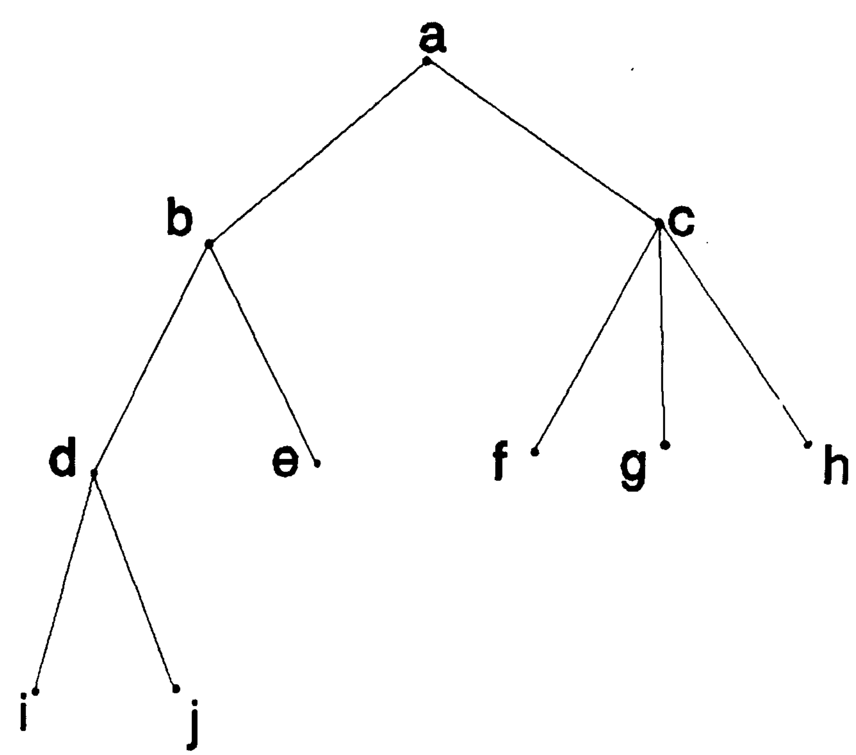


Figure 4.1 A Tree

Theorem 1: Let graph T be a tree. Then $\chi(T_c) = \Delta T+1$.

Proof: First, we will show that $\chi(T_c) \leq \Delta T+1$.

Let T be a tree and let v be the vertex of maximum degree d, i.e., $\deg(v)=d=\Delta T$. We may color the vertices $v_1, v_2, \ldots, v_d$ adjacent to v such that $\mathrm{color}(v_i)=i+1$, coloring vertex v with color 1.

We must show that these d+1 colors are sufficient to color T_c.

Suppose, by way of contradiction, that there is some vertex u in T which requires some color d+2. This implies that u is within distance 2 of d+1 vertices. Let $\deg(u)=d'$, where $d' \leq d$. But that means there are at least $(d+1)-d' \geq 1$ vertices of distance exactly 2 from u. Let w be one of those vertices. Now w must also be within distance 2 of those vertices immediately adjacent to u, else we could duplicate the colors used on them. Let u_2 be one such vertex. But as pictured in Figure 4.2, this implies that there is a path of length 2 from u to w via some vertex u_1, a path of length 2 from w to u_2 and a path of length 1 from u to u_2. But this means there is a cycle, which contradicts the assumption that T is a tree. Therefore there cannot be a vertex which requires d+2 colors, and hence $\chi(T_c) \leq \Delta T+1$.

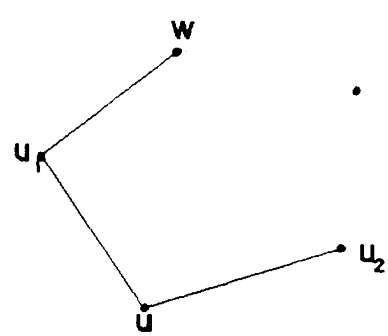


Figure 4.2 A Cycle in T; T cannot be a tree.

Now we must show that $\chi(T_c) \geq \Delta T+1$. Let v be a vertex in T of maximum degree, i.e., $\deg(v)=\Delta T$. Then clearly we must have at least $\Delta T+1$ colors to distinctly color v and its adjacent vertices, hence $\chi(T_c) \geq \Delta T+1$.

But now this implies that $\chi(T_c) = \Delta T+1$. ■

Another specific type of graph that has a closed form chromatic number for its conflict graph is a wheel $W_{1,n}$, as pictured in Figure 4.3.

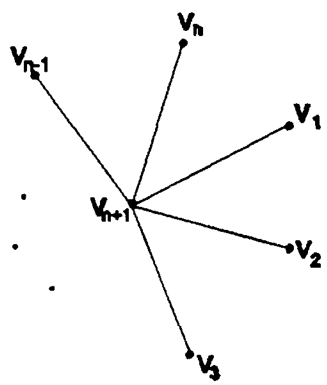


Figure 4.3 A Wheel

<u>Theorem 2</u>: Let G be a wheel $W_{1,n}$. Then $\chi(W_c) = n+1$.

<u>Proof</u>: Let W be a wheel. By the definition of a wheel, we know that there is some vertex v_{n+1} which is adjacent to each of the other n vertices in the graph.

But this implies that each vertex in W is of distance at most two from every other vertex via v_{n+1}, and hence all vertices would be adjacent in the conflict graph of W. Therefore, $\chi(W_c) = n+1$. ■

Similarly, we may show that the conflict graph of a complete bipartite graph $K_{m,n}$ is a complete graph K_{m+n}, and hence $\chi[(K_{m,n})_c] = m+n$.

Theorem 3: Let G be a complete bipartite graph $K_{m,n}$. Then $\chi(K_{m,n\,c}) = m+n$.

Proof: Let G=(V,E) be the complete bipartite graph $K_{m,n}$. We may then partition V into two subsets V_1 and V_2 of sizes m and n, respectively, where the edge set E contains only those edges joining each vertex in V_1 to every vertex in V_2. But that implies that every vertex in V_1 is distance exactly two from every other vertex in V_1, and similarly every vertex in V_2 is distance exactly two from every other vertex in V_2. Hence, in the conflict graph G_c, every vertex in V_1 is adjacent to every other vertex in V and every vertex in V_2 is adjacent to every other vertex in V, so the chromatic number is $\chi(G_c) = |V| = m+n$. ■

We can generalize the above to any graph G of diameter at most two. Indeed, since the diameter is the maximum distance between any two vertices in the graph, it follows that in any graph G=(V,E) whose diameter does not exceed two, all vertices in its conflict graph G_c must be adjacent to one another. Hence the chromatic number is $\chi(G_c) = |V|$.

We now consider chordal graphs. In the general case, we have counterexample to the proposition that the chromatic number is bounded by ΔG+1 or ΔG+2 as shown in Figure 4.4.

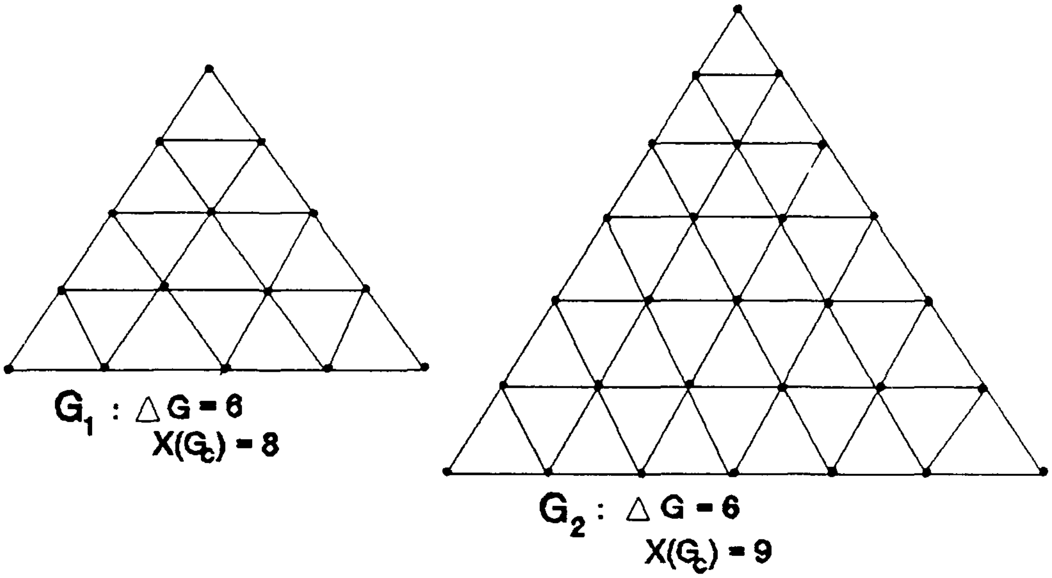


Figure 4.4 Counterexamples to the conjectures of the chromatic number of the conflict graph being max degree + 1 or max degree + 2.

We can, however, obtain such results in the case of graphs that are both chordal and diamond-free. As previously discussed, diamond-free chordal graphs are diamond-free graphs in which every cycle of length strictly greater than three possesses an edge joining two nonconsecutive vertices of the cycle.

We now propose that for a graph G which is both diamond-free and chordal, $\chi(G_c)=\Delta G+1$.

Before going into the required proofs, we will first define an additional term and introduce a theorem.

Definition: A perfect graph $G=(V,E)$ is one in which $\chi(G_A)=\omega(G_A)$, for every subset A of V. [Ref. 7: p. 52]

Theorem 4: Every chordal graph is perfect. [Ref. 7: p. 95]

We propose the following two lemmas.

Lemma 1: If G is a chordal graph, then G_c must also be a chordal graph.

Proof: Let G be a chordal graph and assume, by way of contradiction, that G_c contains an induced n-cycle, $n\geq4$. Because G_c contains G, we know that either G contained the induced n-cycle or the induced n-cycle was constucted from G in its conflict graph.

Consider the first case. But G is given to be a chordal graph, hence cannot contain an induced n-cycle where $n\geq4$.

But this implies that the second case must hold, that is that the induced n-cycle must have been constructed from G in its conflict graph. Consider the n-cycle, x_1, x_2, ..., x_i, x_{i+1}, ..., x_n, x_1, given in Figure 4.5.

Now, for it to have been constructed from G, there must have been at least one edge $\{x_i, x_{i+1}\}$ in $E(G_c)$ that was not in $E(G)$. But this implies that there was some vertex y not in the given n-cycle such that $\{x_i, y\}$ and $\{x_{i+1}, y\}$ were in $E(G)$. But this means that there was an induced cycle of length at least n+1 in G, a contradiction. Therefore, given that G is a chordal graph, G_c cannot contain an induced n-cycle and must therefore also be a chordal graph. ■

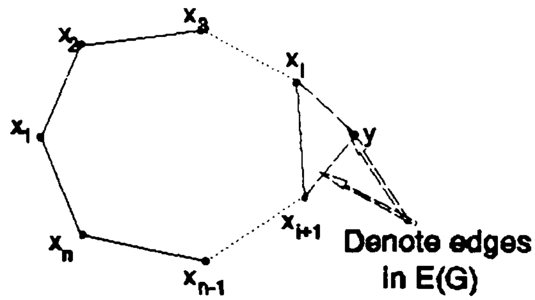


Figure 4.5 The (n+1)-cycle in G; A Contradiction

Lemma 2: If G is a diamond-free chordal graph with maximum degree ΔG, then $\omega(G_c)=\Delta G+1$.

Proof: We must first show that $\omega(G_c)\geq\Delta G+1$, then show $i\omega(G_c)\leq\Delta G+1$.

Now, since ΔG is the maximum degree of G, we know that for some vertex v such that $\deg(v)=\Delta G$, there exists a clique of size $\Delta G+1$ in G_c centered at v. Therefore we know that $\omega(G_c)\geq\Delta G+1$.

We now show that $\omega(G_c)\leq\Delta G+1$. Suppose, by way of contradiction, there is a clique of size $\Delta G+2$ in G_c and let u be a vertex in that clique with degree $d\leq\Delta G$. Then there is at least one vertex, say w, of distance exactly two from u.

Now w must be adjacent to exactly one of those vertices, say x, which is also adjacent to u, otherwise we would create either a diamond or a cycle of length four or greater.

Consider w. Now every vertex in the clique that is adjacent to w must also be adjacent to x, else it would be in a cycle of length greater than three or would not be within distance two of u. Similarly, every vertex in the clique that is adjacent to u must also be adjacent to x, else it would be in a cycle of length greater than three or would not be within distance two of w. But this implies that every vertex in the clique must be adjacent to x, which in turn implies that $deg(x)=\Delta G+1$, a contradiction. Therefore, there cannot exist a clique of size $\Delta G+2$ and $\omega(G_c)=\Delta G+1$. ■

We can now combine the preceding results to prove the following theorem.

<u>Theorem 5</u>: If G is a diamond-free chordal graph, then $\chi(G_c)=\omega(G_c)=\Delta G+1$.

<u>Proof</u>: Let G be a diamond-free chordal graph with maximum degree ΔG. Then by Lemma 1, we know that G_c is a chordal graph and hence is perfect. But this implies that $\chi(G_c')=\omega(G_c')$ for every subgraph G_c' of G_c, and indeed G_c itself. Now, by Lemma 2, $\omega(G_c)=\Delta G+1$, and therefore we have that for a diamond-free chordal graph G, $\chi(G_c)=\omega(G_c)=\Delta G+1$. ■

B. GENERAL GRAPHS AND THEIR CONFLICT GRAPHS

It was previously believed that it would be possible to find a closed form upper bound on the chromatic number of a conflict graph for a general graph G. Indeed it was believed that this upper bound would be of the form $\Delta G + c$, where ΔG is defined to be the maximum degree of all vertices in G and c is a fixed positive integer. Although counterexamples have not been found for every possible value of c between 0 and $|V|-\Delta G$, for specific values of c, counterexamples have been found. For example, given the conjecture that $\chi(G_c) \leq \Delta G+2$, the 4-regular graph on 17 vertices in Figure 4.6 provides a counterexample. Indeed, while $\Delta G = 4$ and $\chi(G) = 5$, it can be shown that $\chi(G_c)=9$.

In attempting to analyze the characteristics of graphs that lend some degree of understanding to how the chromatic numbers of their conflict graphs may be inferred, degree sequences, maximum degrees, eccentricities of vertices, diameters and radii, and subgraphs of numerous graphs were observed.

One case of interest was the example of two similar 3-regular graphs on 10 vertices as displayed in Figure 4.7.

In both G_1 and G_2, their respective diameters and radii were 3. While these characteristics are identical, and given $\Delta G_1 = \Delta G_2 = 3$, we find, however, that $\chi[(G_1)_c] = 6$ and

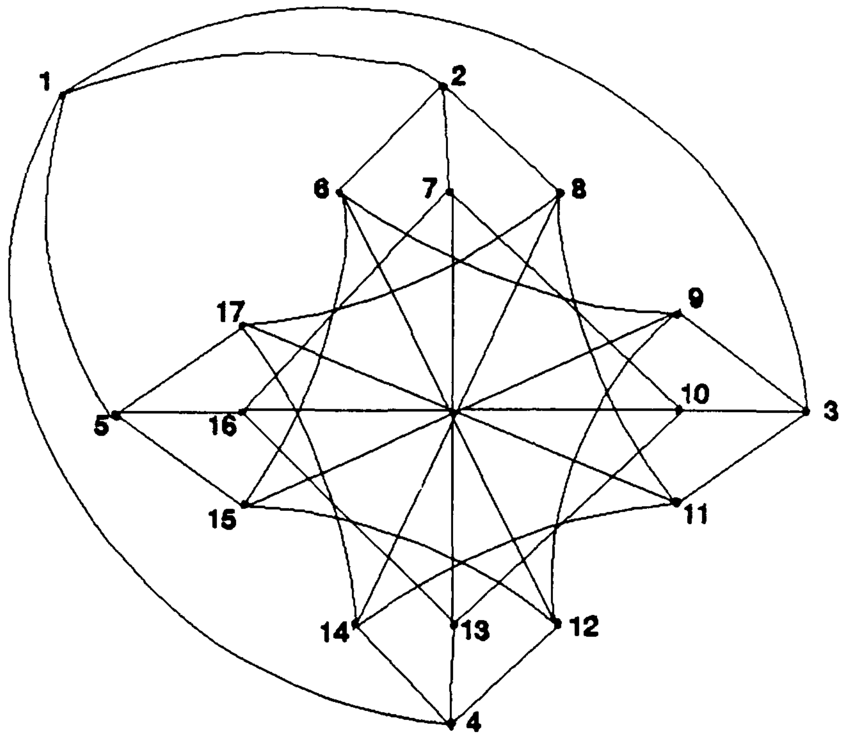


Figure 4.6 A counterexample to an upper bound of max degree + 2

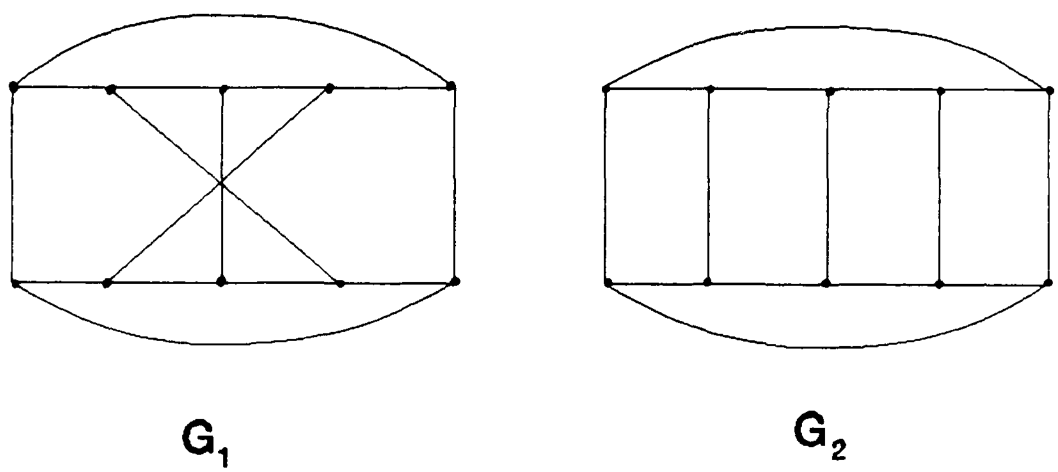


Figure 4.7 Two Similar 3-Regular Graphs

$\chi[(G_2)_c] = 5$. An indication of why this disparity exists might be that the minimum length cycle in G_1 is 5, while the minimum length cycle in G_2 is 4.

This leads us to consider whether a proper subgraph of a graph G can determine the chromatic number of its conflict graph G_c. An immediate class of counterexamples is the class of graphs with diameter at most two, where the conflict graph is a complete graph and hence requires all $|V|$ vertices in order to create a full chromatic number coloring. An example of such a graph is shown in Figure 4.8, where the graph G has 5 vertices and diameter 2, and its conflict graph G_c is the complete graph K_5 with chromatic number $\chi(G_c) = 5$. Thus, by definition, every proper subgraph has at most 4 vertices and cannot yield the full chromatic number coloring of 5.

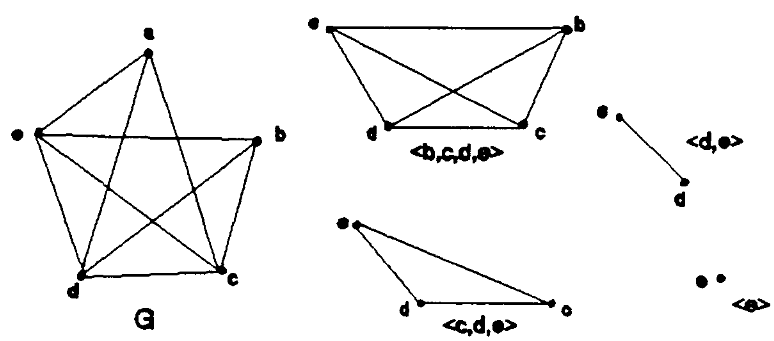


Figure 4.8 A Graph G and Its Proper Subgraphs

C. AN ALGORITHM FOR THE CHROMATIC NUMBER OF A CONFLICT GRAPH

Since we have shown that there is no general solution by which to define the relationship between the structure of a

general graph and the chromatic number of its conflict graph, and since generally acceptable coloring algorithms cannot guarantee optimal solutions in polynomial time, we must accept that less than optimal results may instead be obtainable and more cost beneficial. We have therefore developed an algorithm which runs in polynomial time to find an upper bound on the desired chromatic number. In the algorithm, starting vertices are scanned in degree order and the vertices within distance two and three are "colored" according to the lowest possible coloring for that particular subgraph centered at the starting vertex. We then "mark" those vertices that were central in the subgraph, i.e., within distance two of the starting vertex, and continue with the next unmarked starting vertex. This procedure is followed until all vertices are marked. The assumption of this algorithm is that the chromatic number is a function of the largest area of influence from some central vertex, and not necessarily a direct function of the largest induced clique size of the conflict graph.

Algorithm

Input: A graph G=(V,E)

Output: An upper bound on the chromatic number of the conflict graph $G_c=(V,E_c)$

Step 1: (label vertices by degree) Label the vertices $v_1, v_2, \ldots, v_n$ such that $deg(v_1) >= deg(v_2) >= \ldots >= deg(v_n)$. (Ties can be broken arbitrarily.)

Step 2: Assign color 1 to the first unmarked vertex in the list. This will be the starting vertex. Assign to Set Q the starting vertex and all those vertices adjacent to the starting vertex, and color each uncolored vertex in Q with a consecutive color number.

Step 3: Assign to Set I all those vertices within distance one of some member of the Set Q. Scan each member v_i of I in the given order and assign to it the lowest possible used color j if there is a vertex in Q or I assigned the color j which is of distance three or greater from v_i, and such that no two vertices in I have the same color number if they have an adjacent vertex in common. If there exists no such vertex in Q or I which will allow the duplicate coloring of v_i, then assign to v_i the next available color number. Continue until all members of I are colored.

Step 4: Assign to Set J all those vertices within distance one of some member of the Set I. Scan each member v_j of J in the given order and assign to it the color 1 if there is no other element of J which has been assigned the color 1 and is within distance two of v_j. If there is such an element already assigned the color 1, then if there is a vertex in Q or I assigned the color j which is of distance three or greater from v_j, such that no adjacent vertex in I has that same color, and finally that no two vertices in J have the same color number if they have an adjacent vertex in common, then color v_j with that color. If there exists no such vertex in Q or I which will allow the duplicate coloring of v_j, then assign to v_j the next available color number. Continue until all members of J are colored.

Step 5: Mark all vertices in Set Q and Set I. Denote the starting vertex and the last color number attained in Steps 2, 3 and 4.

Step 6: If any unmarked vertices remain, reinitialize the sets Q and I to the empty set. Go to Step 2.

Step 7: Select the highest coloring number obtained and the starting vertex associated with it. This is the upper bound on the chromatic number for the conflict graph associated with the given graph.

D. EXAMPLES

1. Example 1.

Using the graph from Figure 4.6, we will use the above algorithm to find the upper bound on the chromatic number of its conflict graph. Figure 4.9 shows the graph with its vertices numbered consecutively.

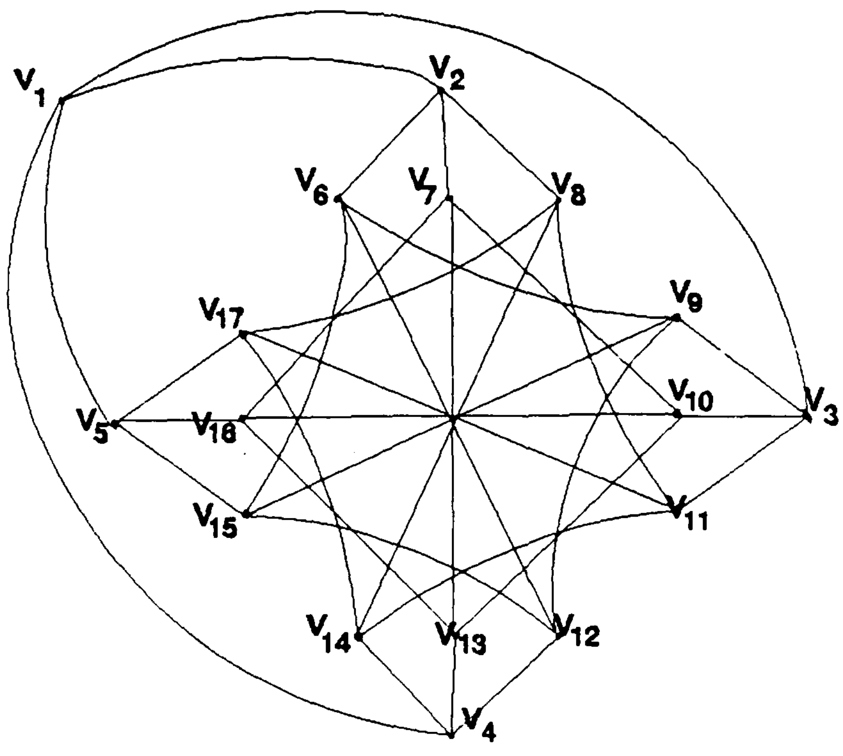


Figure 4.9 Example 1

Step 1: (4-regular graph - present numbering satisfactory)

Step 2: v_1 - color 1;$Q=\{v_1,v_2,v_3,v_4,v_5\}$

v_2 - color 2; v_3 - color 3; v_4 - color 4;

v_5 - color 5

Step 3: $I=\{v_6,v_7,...,v_{16},v_{17}\}$

v_6,v_{10},v_{14} - color 6; v_7,v_9,v_{17} - color 7;

v_8,v_{12},v_{16} - color 8; v_{11},v_{13},v_{15} - color 9

Step 4: $J=\{\}$

Step 5: Marked = $\{v_1,v_2,...,v_{16},v_{17}\}$

Starting vertex v_1 gave upper bound of 9

Step 6: No unmarked vertices remain.

Step 7: $\chi(G_c)\leq 9$

2. Example 2.

Using graph G_1 originally shown in Figure 4.7, we will use the algorithm to find the upper bound on the chromatic number of its conflict graph. Figure 4.10 shows the graph with its vertices numbered consecutively.

Step 1: (3-regular graph - present numbering satisfactory)

Step 2: v_1 - color 1;$Q=\{v_1,v_2,v_5,v_6\}$

v_2 - color 2; v_5 - color 3; v_6 - color 4

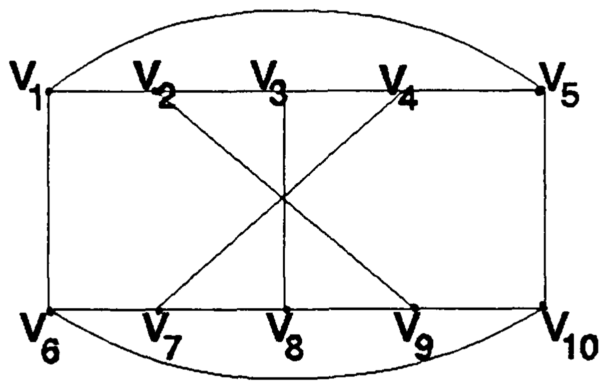
Step 3: $I=\{v_3,v_4,v_7,v_{10}\}$;v_3 - color 4; v_4 - color 5;

v_7 - color 2; v_{10} - color 6

Step 4: $J=\{v_8,v_9\}$; v_8 - color 1; v_9 - color 3

Step 5: Marked=$\{v_1, v_2, \ldots, v_6, v_7, v_{10}\}$

Starting vertex v_1 gave upper bound of 6



G_1

Figure 4.10 Example 2

Step 6: v_8 and v_9 remain unmarked; go to Step 2

Step 2: v_8 - color 1; Q=$\{v_3, v_7, v_8, v_9\}$

v_3 - color 2; v_7 - color 3; v_9 - color 4

Step 3: I=$\{v_2, v_4, v_6, v_{10}\}$; v_2 - color 3; v_4 - color 4;

v_6 - color 2; v_{10} - color 5

Step 4: J=$\{v_1, v_5\}$; v_1 - color 1; v_5 - color 6

Step 5: Marked=$\{v_1, v_2, \ldots, v_9, v_{10}\}$

Starting vertex v_8 gave upper bound 6

Step 6: No unmarked vertices remain.

Step 7: $\chi(G_c) \leq 6$

E. PROOF AND ANALYSIS OF THE ALGORITHM

Proof: Given a connected undirected graph $G=(V,E)$, suppose that the algorithm produced an upper bound of B for the chromatic number of the conflict graph G_c around the vertex v_s. Now, by way of contradiction, suppose that the algorithm did not work, i,e., that there is some vertex v_m that was marked during the coloring, but whose neighborhood required B+1 colors.

Now, since v_m was marked in error, either v_m was adjacent to v_s or v_m was distance two from v_s.

case i: v_m was adjacent to v_s.

Since v_m requires B+1 colors, this means that v_m is within distance two of B other vertices. But the algorithm already scanned distance three from v_s, and this would have encompassed everything within distance two of v_m, hence the additional color would have been picked up had v_m been adjacent to v_s.

case ii: v_m was distance two from v_s.

Again, since v_m requires B+1 colors, this means that v_m is within distance two of B other vertices, all of which must be within two of each other else some color could be repeated among them.

Now, since v_s is within distance two of v_m, by assumption, this implies that v_s is within distance two of the B-1 other vertices near v_m, as well as v_m itself, all of which are colored distinctly. But this further implies that the coloring around v_s must also be B+1, a contradiction.

Therefore, since we have shown that the cases under which v_m could have been overlooked could not have occurred, the algorithm must have worked, and we are done. ■

In analyzing the algorithm we first look at the requirements for Step 1, the sorting of vertices in descending degree order. Such a sort can be done in $O(n\log n)$ time. Steps 2 through 6 are then conducted at most n times, during which each vertex in the graph will be scanned and possibly marked at most once. This implies that there are at most $n\log n+n^2$ operations during the entire algorithm, thus the algorithm is polynomial of $O(n^2)$.

V. SUMMARY AND CONCLUSIONS

We have found that Naval Communication Networks can be easily modelled in graphical form with ships in the network represented by vertices and their capabilities to communicate between ships represented by edges between the vertices. By doing so, we were then able to look at optimal transmission time slot utilization by assigning ships capable of conflicting transmissions different time slots (or colors).

This optimization problem became one of analyzing the structure of the graph representing the given network and finding the chromatic number of its conflict graph. While an optimal solution is nearly impossible to find in all cases, an upper bound on the desired value can be found in most cases, and indeed in specific types of graphs an exact solution is possible.

This research has been directed towards finding those graphs in which an exact solution is possible, while providing an algorithm by which to find an upper bound on the solution when a general graph is encountered. This is not an exhaustive review of all possible types of graphs, but merely a sampling of the more obvious forms.

In using these results to solve the slot utilization problem, the analyst is able to model the Battle Group as a graph, determine if possible the class of graph it most

closely resembles, and in some cases obtain an immediate result for the number of time slots needed in an optimal transmission cycle. If desired, further analysis of the graph, possibly by running a coloring algorithm on the conflict graph, could then provide specific slot assignments.

In less than desirable situations where the graphical model is of a general form, the algorithm described in Chapter IV could be used to find an upper bound on the number of slots needed in the transmission cycle. While this may not be optimal, it should mark some improvement over the current practice of assigning one slot series per ship. If desired, specific slot assignments could also be made using a coloring algorithm on the conflict graph.

LIST OF REFERENCES

1. Shared Adaptive Internetworking Technology: Project Description (RC32A13), Naval Ocean Systems Center

2. Brualdi, R.A., Introductory Combinatorics, 2nd Edition, North Holland Publishing, New York

3. Roberts, F.S., Applied Combinatorics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984

4. Buckley, F., and Harary, F., Distance in Graphs, Addison-Wesley Publishing Company, Redwood City, California, 1990

5. Gould, R., Graph Theory, Benjamin/Cummings Publishing Company, Inc., Menlo Park, California, 1988

6. Roberts, F.S., Discrete Mathematical Models with Applications to Social, Biological and Environmental Problems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976 (pp113-129)

7. Golumbic, M.C., Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980

8. Lundgren, J.R., and Rasmussen, C.W., "Interval Graphs with Noninterval Competition Graphs", to appear in Congressus Numerantium

9. Lundgren, J.R., "Food Webs, Competition Graphs, Competition-Common Enemy Graphs, and Niche Graphs", Applications of Combinatorics and Graph Theory to the Biological and Social Sciences, Springer-Verlag "IMH Volumes in Mathematics and Its Applications, Vol 17, 1989"

10. Raychaudhuri, A., and Roberts, F.S., "Generalized Competition Graphs and Their Applications", in P. Brucker and R. Pauly (eds), Methods of Operations Research, 49, Anton Hais, Konigstein, West Germany, 1985, pp295-311

11. Roberts, F.S., "From Garbage to Rainbows: Generalizations of Graph Coloring and Their Applications", RUTCOR Research Report #36-88, July 1988

12. Manber, Udi., Introduction to Algorithms: A Creative Approach, Addison-Wesley Publishing Company, Menlo Park, California, 1989

13. Garey, M.P., and Johnson, D.S., Computers and Intractiability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Chairman, Code MA Department of Mathematics Naval Postgraduate School Monterey, California 93943	1
4.	Professor C. W. Rasmussen, Code MA/Ra Department of Mathematics Naval Postgraduate School Monterey, California 93943	1
5.	Professor J. R. Thornton, Code MA/Th Department of Mathematics Naval Postgraduate School Monterey, California 93943	1
6.	Professor K. Query 3209 Casa Bonita Drive Bonita, California 91902	1
7.	LT Pamela K. Bell, USN Operations Department Naval Station Mayport, Florida 32228-0112	1